批量正则化 DBN 分类方法研究 *

李蓓蓓，宋　威，戴　鑫

(江南大学 物联网工程学院, 江苏 无锡 214122)

摘　要：针对深度置信网络（DBN）在微调过程中易受训练参数影响的问题，提出一种批量正则化 DBN 分类方法（BNDBN）。该方法首先利用 DBN 进行无监督学习以获得原始数据的高层次表达；然后通过引入尺度变换和平移变换参数对网络中间层的输出特征每一维进行批量正则化处理；并将处理后的特征输入到非线性变换激活层中；最后使用随机梯度下降法对仿射变换参数以及原始网络的参数进行训练学习。BNDBN 方法减少了梯度对参数规模的依赖性，有效解决了因网络参数变化而造成的激活函数值分布变化的问题，提高了训练效率。为了检验所提出方法的有效性，选取 MNIST 手写体数据库和 USPS 手写数字识别库进行测试，通过与 Dropout-DBN、DBN、ANN、SVM、KNN 对比，结果表明，提出的方法分类准确率明显提高，具有更强的特征提取能力。

关键词：深度置信网络；分类；无监督学习；尺度变换；平移变换；批量正则化

中图分类号：TP183

Research on batch regularization DBN classification method

Li Beibei, Song Wei, Dai Xin

(*Institute of Intelligent Systems & Network Computing*, *School of Internet of Things Engineering*, *Jiangnan University*, *Wuxi Jiangsu* 214122, *China*)

Abstract: Aiming at the problem that the deep belief network (DBN) is susceptible to the training parameters during the fine-tune process, this paper proposed a kind of batch normalization DBN classification method (BNDBN) . Firstly, this method used unsupervised learning to obtain high-level representation of raw data. Then through the introduction of scale transformation and translation transformation parameters, it processed the output characteristics of each layer by batch normalization. And it fed the post-processing characteristics into the nonlinear transformation activation layer. Finally, it trained and studied the parameters of the affine transformation and the original network by using the stochastic gradient descent method. The BNDBN method reduced the dependence of the gradient on the parameter size, which effectively resolved the problem of changing the value distribution of activation function caused by the change of network parameters and improves the training efficiency. To verify the effectiveness of the proposed method, it selected MNIST handwritten database and the USPS handwritten digital identification library for testing. Compared with the Dropout-DBN, DBN, ANN, SVM and KNN, the results show that the proposed method significantly improved the classification accuracy and had stronger feature extraction ability.

Key Words: deep belief network; classification; unsupervised learning; scale transformation; translation transformation; batch normalization

0　引言

Hinton 等人于 2006 年提出了深度置信网络(deep belief networks，DBN)以及无监督贪婪逐层训练算法[1]，为解决深度神经网络的优化难题带来了希望。DBN 是一个概率生成模型[2]，通过“逐层初始化” 克服了训练上的难度，它处理高维输入的能力使其成为固有数量维度的任务的理想选择。因 DBN 具有自动学习特征和数据降维[3]的优势，已经成为深度学习应用最广泛的网络结构，目前，DBN 在语音识别[4]、图像分类[5]、人脸识别[6]等相关领域都取得了突破性的进展。

目前，数据集规模日益扩大，更复杂、更深层次的体系结构被提出，使得网络训练变得更加困难，这就需要更有效的训练方式。但仅仅适应庞大的数据集是远远不够的，特别是在深度置信网络中，有监督微调也是不容忽视的一个阶段。随机梯

基金项目：国家自然科学基金资助项目（61673193）；中央高校基本科研业务费资助项目（JUSRP51635B，JUSRP51510）；江苏自然科学基金资助项目（BK20150159）

作者简介：李蓓蓓（1993-），女，山东济宁人，硕士研究生，主要研究方向为深度学习（lbb6112@163.com）；宋威（1981-），男，湖北恩施人，副教授，博士，主要研究方向为数据挖掘、人工智能等；戴鑫（1995-），男，江苏盐城人，本科生，主要研究方向为机器学习.

度下降（SGD）[7]是微调深度置信网络最有效的方法之一，Adagrad[8]和动量[9]等方面已经有了新的改进。文献[10]提出一种高效自然梯度的深层神经网络并行训练算法应用在图像分类中，有效地提高了算法的收敛性；文献[11,12]分别提出了 Dropout 和 DropConnect 算法，其主要目的在于引入稀疏性到网络模型中，减弱了神经元节点间的联合适应性，防止过度拟合；文献[13]则将 Dropout 算法引入到 DBN 的微调过程，以提高网络的泛化能力和分类判别性；文献[14~16]通过元启发式搜索算法及其变种算法来微调 DBN 的参数，以获得近似最优解，避免了陷入局部最优的问题。

在 DBN 的微调阶段，优化目标是最小化给定标签和网络输出之间距离的损失函数，虽然这些算法的改进不同程度的提高了网络训练效果，但仍然对模型超参数敏感，需要更小心的初始化。因为每一层网络的输入都是用所有的下层参数来计算的，所以下层某个参数的微小改变有可能随着网络层数的增加而被逐层放大，那么该层网络需要拟合新的分布，以使网络达到稳定状态，这增加了网络训练的复杂度，降低了网络的训练速度，甚至导致深层网络的效果反而不如浅层网络。

为了解决传统 DBN 训练方法存在的不足，结合批量正则化算法（batch normalization）[17]的优点，本文提出了一种批量正则化 DBN（BNDBN）分类方法。该方法引入了尺度变换和平移变换参数，经过变换重构可以恢复原始网络所学习的特征分布，使得 BNDBN 与 DBN 相比具有稳定的网络性能。通过对 MNIST 和 USPS 手写数字识别库分类的实验分析，与 Dropout-DBN，DBN，人工神经网络(ANN)，支持向量机(SVM)和 K 近邻方法(KNN)算法进行对比，证明了本文方法具有更优的分类准确率。

1　相关工作

1.1　深度置信网络

DBN 是一种概率生成模型，可以看成是由多个受限玻尔兹曼机（restricted boltzmann machine, RBM）[18]堆叠而成的深层网络。RBM 是一种对称连接的随机神经网络，该网络由可见层 v 和隐藏层 h 组成，如图 1 所示。

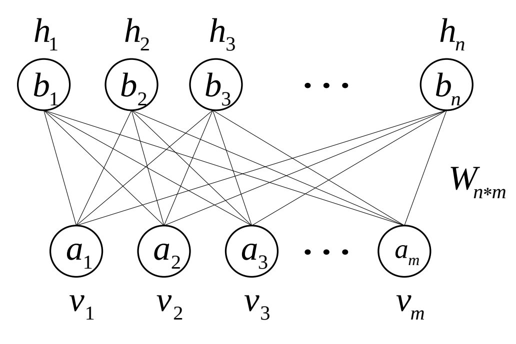


图 1　RBM 网络结构图

RBM 是基于能量的模型，可见层和隐藏层的能量函数公式如下：

$$E(v,h\mid\theta)=-\sum_{i=1}^{n}a_iv_i-\sum_{j=1}^{m}b_jh_j-\sum_{i=1}^{n}\sum_{j=1}^{m}v_iW_{ij}h_j \tag{1}$$

其中：$\theta=\{W_{ij},a_i,b_j\}$。$W_{ij}$ 是可见层和隐藏层之间的权重矩阵，a_i 表示可见层节点的偏置，b_j 表示隐藏层节点的偏置。

该模型产生可见层节点和隐藏层节点的条件概率如下：

$$P(v\mid\theta)=\frac{1}{Z_\theta}\sum\nolimits_h e^{-E(v,h\mid\theta)} \tag{2}$$

$$P(h\mid\theta)=\frac{1}{Z_\theta}\sum\nolimits_v e^{-E(v,h\mid\theta)} \tag{3}$$

当给定可见层节点的状态时，各隐藏层节点之间的激活状态是条件独立的，故第 j 个隐藏层节点的激活状态为:

$$P(h_j=1\mid v)=\sigma(\sum_i W_{ij}v_i+b_j) \tag{4}$$

其中: $\sigma(x)=\dfrac{1}{(1+e^{-x})}$ 为非线性的 sigmoid 激活函数。当给定隐藏层节点的状态时，第 i 个可见层节点的激活状态为

$$P(v_i=1\mid h)=\sigma(\sum_j W_{ij}h_j+a_i) \tag{5}$$

DBN 训练过程包括预训练和微调阶段：根据输入数据对网络进行预训练，选用对比散度（contrastive divergence, CD-k）算法[19]无监督的自底向上单独训练每一层 RBM,得到相应的权值和偏置，最终获得数据的高层次特征；再利用自顶向下的有监督学习-反向传播（back propagation,BP）算法[20]微调整个网络，使 DBN 模型能很好地拟合输入数据。结构如图 2 所示。

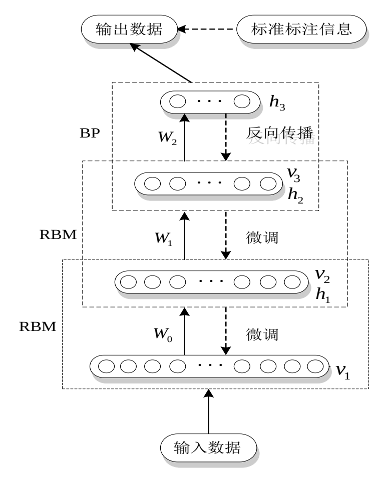


图 2　DBN 训练结构

1.2　Softmax 分类器

如果将深度置信网络应用在分类上，需要在网络的最后一层加入分类器。为了使网络应用更广泛，本文使用 Softmax 分类器来进行分类。

对于训练集 $\{(x^{(1)},y^{(1)}),(x^{(2)},y^{(2)}),\ldots,(x^{(m)},y^{(m)})\}$，标签 y 取 k 个不同的值，表示有 k 个类别。设 $p(y=j\mid x)$ 表示输入 x 的情况下，样本被判为类别 j 的概率。所以对于一个 k 类的分

类器，输出是一个 k 维的向量(向量的元素和为 1)，输出为

$$h_\theta(x^{(i)})=\begin{bmatrix} p(y^{(i)}=1|x^{(i)};\theta) \\ p(y^{(i)}=2|x^{(i)};\theta) \\ \cdots \\ p(y^{(i)}=k|x^{(i)};\theta) \end{bmatrix} =\frac{1}{\sum_{j=1}^{k}\exp(\theta_j^T x^{(i)})}\begin{bmatrix} \exp(\theta_1^T x^{(i)}) \\ \exp(\theta_2^T x^{(i)}) \\ \cdots \\ \exp(\theta_k^T x^{(i)}) \end{bmatrix} \tag{6}$$

其中：θ 是一个矩阵，每一行看做是一个类别所对应分类器的参数，共 k 行。$\frac{1}{\sum_{j=1}^{k}\exp(\theta_j^T x^{(i)})}$ 是对概率分布进行归一化，从而使所有的概率之和为 1。因此，Softmax 分类器的代价函数为

$$J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}1\{y^{(i)}=j\}\log\frac{\exp(\theta_j^T x^{(i)})}{\sum_{l=1}^{k}\exp(\theta_l^T x^{(i)})} \tag{7}$$

其中：1{·}是一个指示性函数，即 1{值为真的表达式}=1，1{值为假的表达式}=0。本文使用梯度下降算法来最小化代价函数求解。

2　基于批量正则化的 DBN 分类方法

2.1　批量正则化算法

由于传统正则化算法不是处处连续求导，批量正则化（batch normalization，BN）[17,21]算法对其进行了改进。首先，传统的正则化算法是对网络中每层的输入特征进行联合正则化，而 BN 算法对网络中每层的每个标量特征进行独立正则化方法处理，并且是对输入的样本分批进行处理。

其次，因为每层网络的输入随着训练参数的改变而改变，导致改变后的输入无法完整表达原有输入特征。对于某一层 k 维的输入数据 x，BN 算法引入了尺度变换参数 γ 和平移变换参数 β。通过变换重构使得数据也会落入非线性分布中，从而保持了模型的表达能力。经过参数的训练学习，当 $\gamma^{(k)}=Var[x^{(k)}]$ 即输入的标准差，$\beta^{(k)}=E[x^{(k)}]$ 即输入的期望值时，可以完全恢复原始网络所要学习的特征分布。

2.2　批量正则化 DBN 结构

在传统 DBN 微调过程中，每层输入分布会随着参数的变化而变化，增加了网络的训练复杂度。如果让非线性输入的分布在训练过程中保持更加稳定的状态，那么优化 DBN 的过程中会降低出现问题的可能性。因此本文在微调阶段中引入批量正则化算法。BNDBN 结构如图 3 所示。

BNDBN 方法可以减少梯度对参数大小或初始值的依赖。传统方法进行迭代更新网络参数时，过高的学习率会导致梯度消失、陷入局部极小值等问题。BNDBN 方法引入尺度变换参数 γ 和平移变换参数 β，对每个隐层的每一维输出特征进行批量正则化处理，通过变换重构将每层的数据分布转变为稳定的标准正态分布，防止参数的微小变化通过深层网络后扩大为梯度的次优变化，使得网络训练时将不会受参数范围的影响。

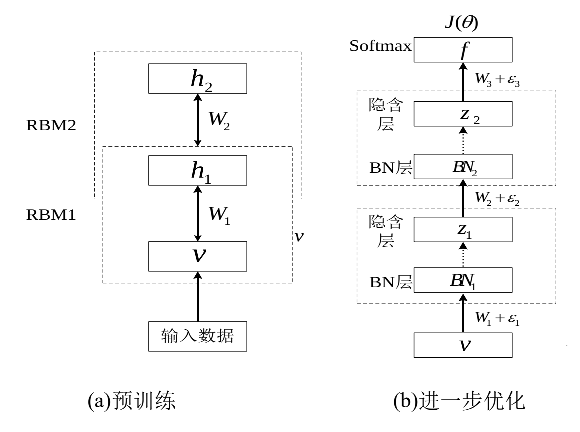


图 3　BNDBN 结构

在图 3(b)中引入批量正则化层即 BN 层，对输入特征进行处理，然后输入到激活函数层。对于某一层具有 k 维的输入 x，每一批样本集合为 $D=\{x_1 \dots x_m\}$，网络的隐藏层层数为 l，每层对输入的每一维都进行正则化，公式如下所示：

$$\mu_D^{(l)}=\frac{1}{m}\sum_{i=1}^{m}x_i^{(l)} \tag{8}$$

$$\sigma_D^{(l)^2}=\frac{1}{m}\sum_{i=1}^{m}\left(x_i^{(l)}-\mu_D^{(l)}\right)^2 \tag{9}$$

$$\hat{x}_i^{(l)}=\frac{x_i^{(l)}-\mu_D^{(l)}}{\sigma_D^{(l)}}=\sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i^{(l)}-\mu_D^{(l)})^2} \tag{10}$$

其中：$x^{(k)}$ 表示输入为 x 的第 k 维，μ_B 表示计算样本集合 D 的均值，σ_B^2 表示计算样本集合 D 的方差，$\hat{x}_i$ 表示对输入 x 的每一维进行正则化处理。

通过尺度变换和平移变换参数来保持模型的表达能力，变换后的公式如下：

$$y^{(l)}=\gamma^{(l)}\sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i^{(l)}-\mu_D^{(l)})^2}+\beta^{(l)}\equiv BN_{\gamma,\beta}(x_i^{(l)}) \tag{11}$$

其中：$y^{(k)}$ 是对 $\hat{x}^{(k)}$ 批量正则化处理后的输出，然后输入到下一个非线性变换激活层中。$\gamma^{(k)}$ 和 $\beta^{(k)}$ 与网络中原有参数一起训练学习，使用随机梯度下降法计算梯度，不断迭代更新。

DBN 网络每个隐含层最后的输出计算如下：

$$\begin{aligned} z^{(l+1)} &= f(W(l)y^{(l)}+b(l)) \\ &= f(W^{(l)}(\gamma^{(l)}\sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i^{(l)}-\mu_B)^2}+\beta^{(l)})+b^{(l)}) \end{aligned} \tag{12}$$

其中：$f(\cdot)=\frac{1}{(1+e^{-x})}$ 为 sigmoid 函数，权值 W 和偏置 b 是要学习的层参数。经过批量正则化处理的 DBN 需要使用反向传播算法计算代价函数梯度，同时计算批量正则化算法中引入的参数。

批量正则化处理后的式(12)可以用下式代替：

$$z = f(BN(Wu)) \tag{13}$$

其中的 BN 变换单独作用于输入 $x = Wu$ 的每一维分量中。在变换输入中加入参数为 a 的尺度变换后，对批量正则化处理 $BN(Wu) = BN((aW)u)$ 进行求偏导得到如下公式：

$$\frac{\partial BN((aW)u)}{\partial u} = \frac{\partial BN(Wu)}{\partial u} \tag{14}$$

$$\frac{\partial BN((aW)u)}{\partial aW} = \frac{1}{a} \cdot \frac{\partial BN(Wu)}{\partial W} \tag{15}$$

从式(14)和(15)中可以看出，在网络的某一层加入参数为 a 的尺度变换后，并没有影响梯度传播。另外，较大的权重加入尺度变换以后导致梯度减小，从而使得引入的批量正则化算法可以保持参数训练时的稳定性。

2.3　批量正则化 DBN 训练过程

批量正则化 DBN 具体训练过程如下：

a)数据预处理。将图片进行灰度转换，并把灰度值归一化到[0,1]；

b)使用 RBM 构建 DBN 网络，初始化可见层和隐藏层的权重矩阵和偏置，设置学习率、迭代次数以及 Mini-Batch 的大小；

c)将预处理后的数据输入到网络输入层中，采用自下而上的无监督学习方法预训练，使用式(4)计算每个 RBM 隐藏层节点激活状态；使用式(5)计算每个 RBM 可见层节点激活状态；反复进行该步骤，并对概率的对数求偏导，最后使用如下公式更新参数空间的权重矩阵和偏置：

$$\Delta W_{ij} = \varepsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon}) \tag{16}$$

$$\Delta a_i = \varepsilon(\langle v_i \rangle_{data} - \langle v_i \rangle_{recon}) \tag{17}$$

$$\Delta b_j = \varepsilon(\langle h_j \rangle_{data} - \langle h_j \rangle_{recon}) \tag{18}$$

其中：ε 为学习率；ΔW_{ij} 是权重的更新值；Δa_i 和 Δb_j 是偏置的更新值；$\langle \bullet \rangle$ 是对数据求期望；$\langle \bullet \rangle_{data}$ 表示重构之前可见层节点 i 与隐藏层节点 j 相乘的值；$\langle \bullet \rangle_{recon}$ 表示重构之后的值，反映重构模型的分布。

d)进一步优化 BNDBN 网络，将预训练得到的参数值即权重和偏置作为该阶段参数的初始值，使用式(6)~(9)对网络的输入进行批量正则化处理，并输入到激活层。然后在网络最顶层加入一层 Softmax 分类器，使用 Mini-Batch SGD 算法最小化代价函数 $J(\theta)$，计算代价函数的梯度，并计算批正则化处理变换中仿射变换参数 γ 和 β。γ 和 β 求导公式使用链式法则如下：

$$\frac{\partial J(\theta)}{\partial \hat{x}_i} = \frac{\partial J(\theta)}{\partial y_i} \cdot \gamma \tag{19}$$

$$\frac{\partial J(\theta)}{\partial \sigma_D^2} = \sum_{i=1}^{m} \frac{\partial J(\theta)}{\partial \hat{x}_i} \cdot (x_i - \mu_D) \cdot (-\frac{1}{2}) \cdot (\sigma_D^2 + \varepsilon)^{-3/2} \tag{20}$$

$$\frac{\partial J(\theta)}{\partial \mu_D} = (\sum_{i=1}^{m} \frac{\partial J(\theta)}{\partial \hat{x}_i} \cdot \frac{-1}{\sqrt{\sigma^2 + \varepsilon}}) + \frac{\partial J(\theta)}{\partial \sigma_D^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_D)}{m} \tag{21}$$

$$\frac{\partial J(\theta)}{\partial x_i} = \frac{\partial J(\theta)}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma^2 + \varepsilon}} + \frac{\partial J(\theta)}{\partial \sigma_D^2} \cdot \frac{2(x_i - \mu_D)}{m} + \frac{\partial J(\theta)}{\partial \mu_D} \cdot \frac{1}{m} \tag{22}$$

$$\frac{\partial J(\theta)}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial J(\theta)}{\partial y_i} \cdot \hat{x}_i \tag{23}$$

$$\frac{\partial J(\theta)}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial J(\theta)}{\partial y_i} \tag{24}$$

再按照梯度求导公式对 W 和 b 更新。

e)将测试数据输入训练好的网络中进行测试。在批量正则化层使用训练阶段中的标准差的期望和均值的期望来对测试数据进行处理，如下式所示：

$$Var[x] \leftarrow \frac{m}{m-1} E_B[\sigma_B^2] \tag{25}$$

$$\hat{x} = \frac{x - E[x]}{\sqrt{Var[x] + \varepsilon}} \tag{26}$$

$$y = \frac{\gamma}{\sqrt{Var[x] + \varepsilon}} \cdot x + \left(\beta - \frac{\gamma E[x]}{\sqrt{Var[x] + \varepsilon}}\right) \tag{27}$$

其中: $E[x]$ 表示全部批次均值的期望值，$Var[x]$ 表示每个批次标准差的无偏估计。

3　实验结果及分析

3.1　实验数据集

为了验证本文所提出算法的有效性，在 MNIST 和 USPS 手写体数据集上分别进行实验分析。MNIST 手写体数据集包括由 0~9 阿拉伯数字组成的 60 000 个训练样本集和 10 000 个测试样本集，每个图像为 28×28 的像素。从中随机选取 6 000 个作为训练集，2 000 个作为测试集。USPS 数据集是美国邮政服务手写数字识别库，包括 9 298 个手写数字图像，均为 16*16 像素的灰度值，本文对其灰度值作了归一化处理，选取 7 000 个为训练数据，4 000 个为测试数据。

3.2　实验结果分析

3.2.1 参数分析

实验在 Windows7 操作系统上运行，使用 MATLABR 2008b 开发环境。实验参数的选择是进行大量的实验之后选取的较为理想的参数值。若不计入 BN 层，本文选取网络节点数分别为 784-100-100-10 共四层，mini-batch 设为 100，动量参数为 0.9，初始迭代次数设为 100。

为了测试学习率对 BNDBN 的影响，本文在其他参数固定的情况下，改变学习率在区间[0.0005,0.05]内变化。在 MNIST 和 USPS 数据集上进行对比实验分析，如图 4 和 5 所示。

观察图 4 可知，BNDBN 算法具有很好的分类性能。从图 4 中可以看出 BNDBN 的分类错误率曲线都在 Dropout-DBN 和 DBN 的下方，分类错误率都在 6%以下。同时，本文算法比 Dropout-DBN 最优时的分类错误率还低了 0.6%。说明本文算法

具有更好的稳定性，可以使 BNDBN 模型很好地提取到数据的主要特征。

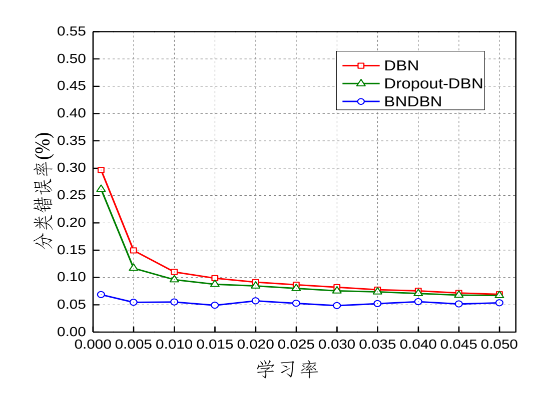


图 4　不同学习率下三种算法的分类错误率对比(MNIST)

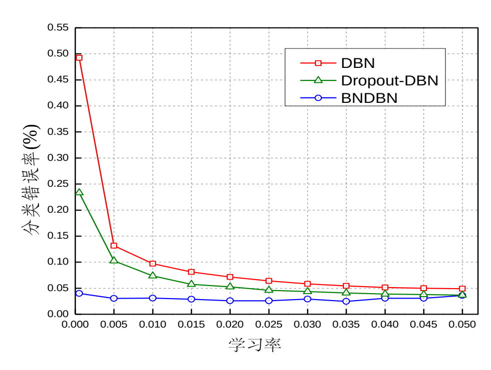


图 5　不同学习率下三种算法的分类错误率对比(USPS)

图 5 显示了在 USPS 数据集中三种算法随着学习率变化的分类错误率对比结果，不难发现， BNDBN 的分类错误率低于其他算法，进一步增加学习率，虽然导致模型最初的训练有点慢，但分类效果比较好。也证明了本文提出的 BNDBN 在保证算法不发散的情况下进一步提高了网络的分类性能。

传统 DBN 的贪婪无监督训练方法可以有效地提取数据特征，各层的权值和偏置会处于最优的位置，使用 SGD 算法进一步训练更容易使网络收敛到最优。因此前一阶段参数的训练效果会影响到微调阶段的学习，进而影响最终分类效果。为了测试 BNDBN 在预训练阶段所需最优迭代次数，本文进行了实验对比，如图 6 和 7 所示。

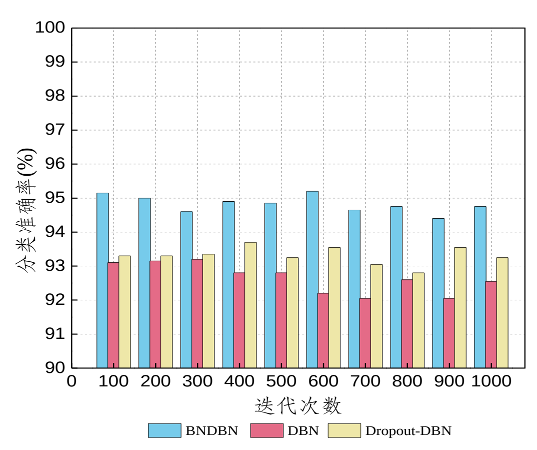


图 6　不同迭代次数下三种算法的分类准确率对比(MNIST)

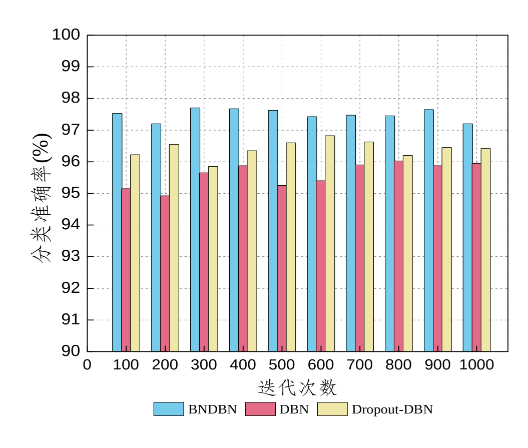


图 7　不同迭代次数下三种算法的分类准确率对比(USPS)

由图 6 和 7 可以看出，在 MNIST 和 USPS 中，训练每个 RBM 迭代的次数对整个网络的性能是有一定影响的，BNDBN 的分类结果都要高于 DBN。在 MNIST 中，迭代次数为 600 次时， BNDBN 的性能最优，准确率达到 95.20%，Dropout-DBN 和 DBN 达到最好的分类结果分别为 93.7%和 93.15%。对于 USPS 数据集，BNDBN 算法在迭代次数为 300 次时，分类准确率达到 97.70%，比 Dropout-DBN 和 DBN 最好时的分类准确率提高了 0.87%和 1.67%。总体来看，与其他算法的对比进一步表明本文算法具有更好分类效果。

为了验证 BNDBN 使用更深层网络的性能，在其他参数固定的情况下，隐含层节点为 100，逐渐增加隐含层的个数，最多增加到 10 层，在 MNIST 和 USPS 上进行分类准确率对比，如图 8 和 9 所示。

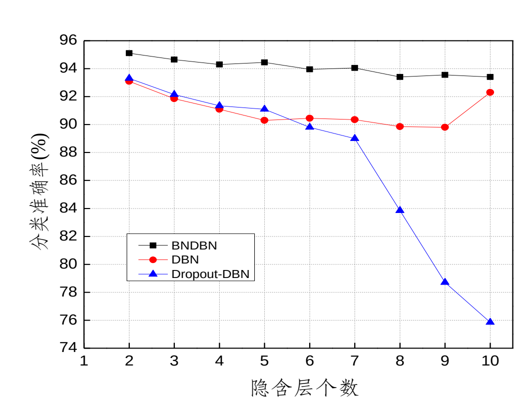


图 8　不同个数隐含层的三种算法的分类准确率对比(MNIST)

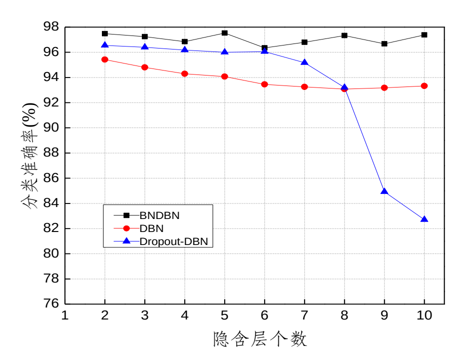


图 9　不同个数隐含层的三种算法的分类准确率对比(USPS)

观察图 8 和 9 可知，开始增加隐含层个数时分类准确率逐渐降低，在 MNIST 数据集上，BNDBN 算法当隐含层个数为 2 时，获得最优的分类准确率；在 USPS 数据集上，当隐含层为 5 时，分类准确率最好；对于 Dropout-DBN 算法，在隐含层个数超过 7 层以后，结果开始变得比较差；但整体来看，本文算法都要优于 DBN 和 Dropout-DBN 算法。

3.2.2 不同算法分类准确率对比

为了进一步验证本文算法的有效性，将本文算法 BNDBN 与 DBN、Dropout-DBN 以及其他应用比较广泛的分类算法即 BP、SVM、KNN 和 DBN 在 MNIST 和 USPS 数据集上进行分类准确率的比较，如表 2 所示。表中所列出的是每种算法最优的分类准确率。其中，ANN 算法中隐含层节点为 100，SVM 采用多项式核函数，KNN 算法采用欧氏距离方法，K 取值为 10。

表 1　六种算法的分类准确率对比(%)

数据集	BNDBN	Dropout-DBN	DBN	ANN	SVM	KNN
MNIST	**95.20**	93.70	93.15	92.6	94.35	93.00
USPS	**97.70**	96.83	96.03	95.33	94.97	93.67

从表 1 可以看出，在 MNIST 和 USPS 中，DBN 的分类结果明显高于 ANN 算法，表明了 DBN 在图像分类任务中的有效性。Dropout-DBN 算法虽然在一定程度上改善了 DBN 存在的过拟合问题，但是仍然受到参数的影响导致分类准确率较低。而本文算法 BNDBN 相对于这五种算法都取得了最优的分类准确率，说明本文算法在一定程度上解决了 DBN 存在的不足，改善了 DBN 的分类效果。

3.2.3 训练时间对比分析

表 2 给出了 BNDBN、Dropout-DBN、 DBN、BP、SVM 和 KNN 算法在 MNIST 和 USPS 数据集上的训练时间对比。从表中可以看出，由于使用深度神经网络的深层结构，计算复杂度比较大，所以相比较于传统分类算法 SVM 和 KNN，训练时间较长，但分类准确率比较高；BNDBN 相对于 DBN 和 Dropout-DBN，训练时间都是最少，而且在较少时间内比 Dropout-DBN 和 DBN 的最好分类准确率分别提高 1.5%和 2%左右；说明本文算法 BNDBN 在训练时间上仍有进一步的提升，加快了训练的收敛速度，具有较优的分类性能。

表 2　不同算法的训练时间对比/min

数据集	BNDBN	Dropout-DBN	DBN	ANN	SVM	KNN
MNIST	16.86	20.61	37.62	5.00	0.18	5.57
USPS	5.31	7.93	8.47	4.04	0.19	3.09

4　结束语

本文提了一种批量正则化 DBN 分类方法，并将它应用到分类识别中。首先介绍了深度置信网络(DBN)的结构，然后详细描述了批量正则化 DBN 分类方法及其训练过程，该方法不仅利用了 DBN 的优异的特征提取能力，同时基于批量正则化算法的优点，解决了传统 DBN 训练方法存在的问题。最后在 MNIST 和 USPS 数据集上进行了对比实验，证明了本文算法具有良好的分类效果。在以后的工作中继续侧重于网络参数优化方面的问题，进一步提升算法的特征提取能力。

参考文献：

[1] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets [J]. Neural Computation, 2006, 18 (7): 1527-1554.

[2] 李慧. 深度生成模型学习算法研究与应用 [D]. 北京：中国科学院大学, 2014.

[3] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks [J]. Science, 2006, 313 (5786): 504-507.

[4] Zhang J, Tao Z Y. Recognition of speech based on deep learning [J]. Electronic Design Engineering, 2015, 23 (18): 72-73.

[5] Liu F, Jiao L, Hou B, et al. POL-SAR image classification based on Wishart DBN and local spatial information [J]. IEEE Trans on Geoscience & Remote Sensing, 2016, 54 (6): 3292-3308.

[6] 林妙真. 基于深度学习的人脸识别研究 [D]. 大连：大连理工大学, 2013.

[7] Bottou L. Large-Scale Machine Learning with Stochastic Gradient Descent [C]// Proc of COMPSTAT. Physica-Verlag, 2010: 177-186.

[8] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization [J]. Journal of Machine Learning Research, 2011, 12 (7): 257-269.

[9] Sutskever I, Martens J, Dahl G, et al. On the importance of initialization and momentum in deep learning [C]// Proc of International Conference on International Conference on Machine Learning. 2013: III-1139.

[10] Povey D, Zhang X, Khudanpur S. Parallel training of deep neural networks with natural gradient and parameter averaging [J]. Eprint Arxiv: 1410. 7455, 2014: 124-145.

[11] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15 (1): 1929-1958.

[12] Wan L, Zeiler M D, Zhang S, et al. Regularization of neural networks using drop connect [C]// Proc of International Conference on Machine Learning. 2013: 1058-1066.

[13] 王忠民, 王希, 宋辉. 基于随机 Dropout 深度信念网络的移动用户行为识别方法 [J]. 计算机应用研究, 2017, 34 (12) .

[14] Papa J P, Scheirer W, Cox D D. Fine-tuning deep belief networks using harmony search [J]. Applied Soft Computing, 2015, 46 (C): 875-885.

[15] Rosa G, Papa J, Costa K, et al. Learning parameters in deep belief networks through firefly algorithm [M]// Artificial Neural Networks in Pattern Recognition. [S. l.] : Springer International Publishing, 2016: 138-149.

[16] Rodrigues D, Yang X S, Papa J P. Fine-tuning deep belief networks using cuckoo search [M]// Bio-Inspired Computation and Applications in Image Processing. 2016: 47–59.

[17] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift [J]. Computer Science, 2015: 448-456.
[18] Fischer A, Igel C. An introduction to restricted Boltzmann machines [M]// Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. Berlin: Springer, 2012: 14-36.
[19] Hinton G E. Training products of experts by minimizing contrastive divergence [J]. Neural Computation, 2002, 14 (8): 1771-1800.
[20] Vogl T P, Mangis J K, Rigler A K, et al. Accelerating the convergence of the back-propagation method [J]. Biological Cybernetics, 1988, 59 (4): 257-263.
[21] Cooijmans T, Ballas N, Laurent C, et al. Recurrent batch normalization [J]. arXiv preprint arXiv: 1603. 09025, 2016.